

Amendments to the Claims:

1 (canceled)

2 (withdrawn): The method of claim 4, wherein the block copy command is associated with a defragmentation operation.

3 (canceled)

4 (withdrawn): A method comprising: receiving a block copy command that is associated with a file system; passing down, by the file system to a differential snapshotter that keeps a translation table and wherein the snapshotter keeps a bitmap of blocks in the volume, wherein, for a bit in the bitmap corresponding to a block in the volume, a first bit value indicates that the snapshotter does not need to take any action when the block is written, and a second bit value indicates that the snapshotter must protect the block with a copy-on-write before the block is written; and wherein the block copy command results in the block move from the location A on the volume to the location B on the volume.

5 (withdrawn): The method of claim 4, further comprising, when a bit in the bitmap corresponding to the location B has the second bit value:

performing a copy-on-write of the location B to a differential area location before the location B is written by the block move;

adding a table entry registering the copy-on-write of the location B to the differential area location; and

changing the bit corresponding to the location B to the first bit value.

6 (withdrawn): The method of claim 4, further comprising, when a bit in the bitmap corresponding to the location B has the first bit value, and when a bit in the bitmap corresponding to the location A has the second bit value:

letting the block move happen;

changing the bit corresponding to the location A to the first bit value;

changing the bit corresponding to the location B to the second bit value; and adding a table entry mapping the location A to the location B.

7 (withdrawn): The method of claim 6, further comprising:

adding a table entry mapping the location B back to the location A.

8 (withdrawn): The method of claim 6, further comprising, when, at a later time, a write of the location B is to occur:

performing a copy-on-write of the location B to a differential area location before the write occurs;

replacing the table entry mapping the location A to the location B with a table entry mapping the location A to the differential area location; and

changing the bit corresponding to the location B to the first bit value.

9 (withdrawn): The method of claim 8, further comprising, when there is a table entry mapping the location B back to the location A, deleting the table entry.

10 (withdrawn): The method of claim 6, further comprising, when a subsequent write of the location B has not yet occurred and a block move from the location B to a location C is initiated:

when a bit in the bitmap corresponding to the location C has the second bit value, performing a copy-on-write of the location C to a differential area location before it is written by the block move;

adding a table entry registering the copy-on-write of the location C to the differential area location; and

changing the bit corresponding to the location C to the first bit value;

otherwise, when the bit corresponding to the location C has the first bit value, letting the block move happen;

changing the bit corresponding to the location B to the first bit value;

changing the bit corresponding to the location C to the second bit value;

replacing the table entry mapping the location A to the location B with a table entry

mapping the location A to the location C; and

when there is a table entry mapping the location B back to the location A, replacing the table entry with a table entry mapping the location C back to the location A.

11 (canceled)

12 (currently amended): In a data storage system comprising a file system and one or more disk volumes, each volume comprising files and blocks, wherein each file is implemented by a set of one or more blocks, wherein blocks currently implementing a file are occupied-space blocks and all other blocks are free-space blocks, a method comprising:

taking a snapshot of an original volume; wherein taking the snapshot includes creating a bitmap comprising a plurality of bit spaces, wherein each bit space corresponds to a respective block in the original volume, wherein each bit space is set either to an ignore value or to a protect value, and wherein initially a bit space is set to the ignore value when the respective block that corresponds to the bit space is a free-space block when the snapshot is taken, and a bit space is set to the protect value when the respective block that corresponds to the bit space is an occupied-space block when the snapshot is taken;

monitoring moves of occupied-space blocks in the original volume, wherein moving an occupied-space block comprises transferring contents from within the occupied-space block to a second block, which is one of the free-space blocks, the second block becoming an occupied-space block, and the first block becoming a free-space block;

monitoring writes of blocks in the original volume; wherein monitoring the writes of the blocks includes determining which blocks to protect by copy-on-write operations before the writing of the blocks and which blocks not to protect by the copy-on-write operations; and producing a snapshot version of blocks in response to a read request.

13 (previously presented): The method of claim 12, wherein monitoring the moves includes intercepting a block copy command passed down by the file system.

14 (original): The method of claim 12 wherein moves of blocks include moves that are associated with a defragmentation operation.

15 (canceled)

16 (previously presented): The method of claim 12, wherein monitoring the moves, monitoring the writes and producing the snapshot version of blocks includes recording information regarding a current actual location for the snapshot version of the blocks in the original volume.

17 (previously presented): The method of claim 16, wherein monitoring a write of a block further comprises:

- when the bit space in the bitmap corresponding to the block is set to the protect value, copying the block to a location in a differential storage space;

- setting the bit space to the ignore value;

- when the block to be written is recorded as the current actual location of one of the snapshot versions that relates to a different block in the original volume, recording the location of the copy in the differential storage space as the current actual location for the snapshot version of that different block, and otherwise recording the location of the copy in the differential storage space as the current actual location for the snapshot version of the block to be written; and

- permitting the write to proceed; and

- when the bit space in the bitmap corresponding to the block is set to the ignore value, permitting the write to proceed.

18 (previously presented): The method of claim 17, further comprising, when the bit space in the bitmap corresponding to the block to be written is set to the protect value:

- deleting each entry in a translation table that indicates that the block to be written is the current actual location for the snapshot version of a different block, and

- recording in a new table entry the location of the copy in the differential storage space as the current actual location for the snapshot version of the block to be written.

19 (currently amended): The method of claim 16, wherein monitoring a move of a first block to a second block comprises:

- when the bit space in the bitmap corresponding to the first block is set to the protect value [[:]] and the bit space in the bitmap corresponding to the second block is set to the

ignore value[[₅]] then:

permitting the move to proceed;

when the first block is recorded as the current actual location of the snapshot version of a third block in the original volume: recording the second block as a new current actual location of the snapshot version of the third block, and otherwise recording the second block as the current actual location of the snapshot version of the first block[[₅]] and exchanging the bits in the bit spaces in the bitmap corresponding to the first block and the second block, so that the bit space corresponding to the first block is set to the ignore value, and the bit space corresponding to the second block is set to the protect value;

~~otherwise,~~

when the bit space corresponding to the second block is set to the protect value: [[₅]]

copying the second block to a location in a differential storage space;

setting the bit space corresponding to the second block to the ignore value;

when the second block is recorded as the current actual location of the snapshot version of a different block in the original volume: [[₅]] recording the location of the copy in the differential storage space as the current actual location of the snapshot version of that different block₁, and otherwise recording the location of the copy in the differential storage space as the current actual location of the snapshot version of the second block; and Q
permitting the move to proceed;

~~otherwise, if~~

when the bit space corresponding to the first block is set to the ignore value,
permitting the move to proceed.

20 (previously presented): The method of claim 19, wherein a translation table is used to record information regarding the current actual location for the snapshot version of the blocks, the method further comprising, when the bit space corresponding to the first block is set to the protect value, if the bit space corresponding to the second block is set to the ignore value, and when the first block is recorded in the table as the current actual location of the snapshot version of a third block in

the original volume:

when the second block and the third block are the same block, deleting each entry in the table that indicates that the first block is the current actual location of the snapshot version of the second block, and

when the second block and the third block are not the same block, recording in a new table entry the second block as the new current actual location of the snapshot version of the third block.

21 (currently amended): The method of claim 16, wherein producing the snapshot version of a blocks in response to a the read request comprises directing the read request to the current actual location of the snapshot version of the blocks.

22 (previously presented): The method of claim 21, wherein a translation table is used to record information regarding the current actual location of the snapshot version of the blocks, the method further comprising:

when the block to be read has a table entry indicating the current actual location of the snapshot version, redirecting the read to that location, and otherwise, directing the read to the version of the block in the original volume.